

# Tipps & Tricks – Nützliche Lösungen zu Problemen und Fragestellungen der SAS Versionen 8, 8.1 und 8.2

**Carina Ortseifen**

Universität Heidelberg  
Universitätsrechenzentrum

Carina.Ortseifen@urz.uni-heidelberg.de

**Grischa Pfister**

Info Ware  
Heidelberg

Grischa\_Pfister@Info-Ware.de

**Heinrich Stürzl**

Dade Behring Marburg  
Statistical Systems & Data Management

Heinrich\_Stuerzl@DadeBehring.com

## Zusammenfassung

Die Versionen 8, 8.1 und 8.2 der SAS Software bieten viele Neuerungen, die sich den Anwendern erst nach und nach offenbaren. Wir möchten in dieser Sitzung einige aus unserer Sicht interessante Themen ansprechen und Hinweise geben, wie Sie sich manches erleichtern können.

1. SAS Output nach Excel exportieren (ODS HTML)
2. Makrovariablen löschen
3. System Optionen auslesen und speichern
4. Externe DLLs in SAS verwenden (nur Windows)
5. SAS Notes oder wie helfe ich mir selbst?
6. PROC UNIVARIATE in V8
7. Die neue Fensterumgebung in V8
  - Enhanced Editor
    - Anpassen der Farbschemata
    - Auto-Text
    - nützliche Tastenkombinationen
  - Explorer
    - Bedienung
    - Eigene Menüeinträge

**Keywords:** Tipps, Tricks, SAS, Version 8, V8, Output, Excel, ODS HTML, Makro, Macro, Makrovariable, Macrovariable, SYMDEL, CALL SYMDEL, Systemoption, System Option, Grafikoption, GOPTION, GETOPTION, Dictionary Table, External DLL, Windows API, KERNEL32.DLL, SASCBTBL, SDK, MODULEN, SAS Notes, Fix, Hotfix, SAS-L, SAS-Ah, Univariate, Histogramm, Probplot, QQ-Plot, ODS RTF, Enhanced Editor, Explorer.

## 1 SAS Output nach Microsoft Excel exportieren (ODS HTML)

Wie lässt sich der Output von SAS Prozeduren in eine Excel Tabelle schreiben, um zum Beispiel ein Datenlisting, das mit PROC PRINT erzeugt wurde mit Excel weiterverarbeiten zu können?

Die hier vorgestellte Methode nutzt dazu ODS HTML und die Tatsache, dass Html-Dateien von Excel direkt gelesen und konvertiert werden können (ab Excel 97). Der Trick dabei: Der HTML-Datei schon bei der Erstellung die Endung **'xls'** geben. Dadurch kann sie direkt in Excel geöffnet und mit File ⇒ Save im Excel Format gespeichert werden.

Dies funktioniert für alle Prozeduren, die Listen-Output erzeugen.

### Beispiel 1.1: PROC PRINT

```
DATA class;
  SET sashelp.class;
  Groesse=height*2.54; * inch --> cm;
  Gewicht=weight*0.45; * lb --> kg;
  FORMAT groesse 3. gewicht 5.1;
  LABEL Groesse="Körpergröße [cm]"
        Gewicht="Körpergewicht [kg]";
RUN;

ODS LISTING CLOSE;
FILENAME report1 'c:\temp\test1.xls'; * Endung '.xls'
  obwohl tatsächlich HTML Datei;
ODS HTML BODY=report1 STYLE=MINIMAL RS=NONE;

TITLE "SAS Output in Excel";
PROC PRINT DATA=class LABEL OBS="Nr"; * Alternatives Label
  für die Spalte 'Obs';
RUN;

ODS HTML CLOSE;
ODS LISTING;
```

Die erzeugte Datei wird in SAS automatisch im Results Viewer ähnlich wie in Excel angezeigt. Dabei wird als Dezimaltrennzeichen der Punkt verwendet, was zu falscher Anzeige (z.B. als Datum) führen kann. Erst beim Öffnen der Datei in Excel (vorher muss sie im im Results Viewer geschlossen werden!) wird das unter Windows eingestellte Dezimaltrennzeichen verwendet, so dass die Daten dort richtig angezeigt werden.

Nr	Name	Sex	Age	Height	Weight	Körpergröße [cm]	Körpergewicht [kg]
1	Alice	F	13	56.5	84.0	144	37.8
2	Barbara	F	13	65.3	98.0	166	44.1
3	Carol	F	14	62.8	102.5	160	46.1
4	Jane	F	12	59.8	84.5	152	38.0
5	Janet	F	15	62.5	112.5	159	50.6
6	Joyce	F	11	51.3	50.5	130	22. Jul
7	Judy	F	14	64.3	90.0	163	40.5
8	Louise	F	12	56.3	77.0	143	34.7
9	Mary	F	15	66.5	112.0	169	50.4

**Hinweise zur ODS HTML Anweisung:**

STYLE=MINIMAL reduziert die HTML Formatierung, was zu einer kleineren Datei und einem schnelleren Import in Excel führt.

Die RS= Option (RECORD\_SEPARATOR) wird nur bei Mainframes benötigt.

**Quelle:**

[www.sas.com/service/techtips/quicktips/pcwrksheet.html](http://www.sas.com/service/techtips/quicktips/pcwrksheet.html)

## 2 Makrovariablen löschen

SAS unterscheidet zwischen globalen und lokalen Makrovariablen. Während lokale Makrovariablen nur zur Laufzeit des Makros existieren, in welchem sie definiert werden, bleiben die globalen Makrovariablen bis zum Ende der SAS Sitzung erhalten.

Ab SAS Version 8.2 gibt es mit %SYMDEL bzw. CALL SYMDEL() erstmals die Möglichkeit, benutzerdefinierte globale Makrovariablen zu löschen.

### Syntax:

```
%SYMDEL makrovar1 <makrovar2 ...>;  
CALL SYMDEL(makrovar1);
```

### Beispiel 2.1:

Einzelne benutzerdefinierte globale Makrovariablen löschen

```
* Globale Makrovariablen definieren;  
%LET SiteNummer=&SYSSITE;  
%LET UserID=&SYSUSERID;  
%LET Betriebssystem=&SYSSCPL;  
%LET SASVersion=&SYSVLONG;  
%PUT _USER_;  
  
* Einzelne globale Makrovariablen löschen;  
%SYMDEL SiteNummer UserID;  
%PUT;  
%PUT _USER_;
```

### Beispiel 2.2:

Makrovariable als Argument von SYMDEL verwenden

```
%LET hilf=Betriebssystem SASVersion;  
%SYMDEL &hilf;  
%PUT;  
%PUT _USER_;
```

**Hinweis:** Dies funktioniert in Version 8.2 nicht, wenn es innerhalb eines Makros verwendet wird (SN-005975), Hotfix verfügbar unter [www.sas.com/techsup/download/hotfix/v82/base/82ba45/82ba45.html](http://www.sas.com/techsup/download/hotfix/v82/base/82ba45/82ba45.html)

### Beispiel 2.3:

Alle benutzerdefinierten globalen Makrovariablen löschen

Dazu die Namen aller globalen Makrovariablen aus der Dictionary Table 'DICTIONARY.MACROS' mit PROC SQL auslesen und als SAS Tabelle (data set) speichern. Jede Makrovariable entspricht darin einer Beobachtung und kann in einem zweiten Datenschnitt gelöscht werden.

Wenn man ohne PROC SQL arbeiten möchte, kann man alternativ auch den SQL View 'SASHELP.VMACRO' benutzen, was allerdings länger dauert.<sup>1</sup>

```
PROC SQL;  
  CREATE TABLE GlobalMakroVars AS  
  SELECT NAME  
  FROM DICTIONARY.MACROS  
  WHERE SCOPE="GLOBAL";  
QUIT;
```

---

<sup>1</sup>Siehe auch Tipps & Tricks der 5. KSFE 2000 in Gießen.

```

* alternativ SQL View 'SASHELP.VMACRO' benutzen (langsamer);
DATA GlobalMakroVars;
  SET SASHELP.VMACRO;
  WHERE SCOPE="GLOBAL";
  KEEP NAME;
RUN;

DATA _NULL_;
  SET GlobalMakroVars;
  CALL SYMDEL(NAME);
/* CALL EXECUTE('%SYMDEL '||NAME||';');*/
RUN;
%PUT;
%PUT _USER_;

```

### 3 Systemoptionen auslesen und speichern

Wie lassen sich die aktuellen Einstellungen der Systemoptionen (system options) oder Grafikoptionen (goptions) von SAS automatisiert auslesen und speichern? Dies ist insbesondere für die Anwendungs- oder Makroentwicklung interessant, wenn Optionen vorübergehend geändert werden und anschließend die ursprünglichen Einstellungen wiederhergestellt werden sollen.

Ab Version 8.2 gibt es die neuen Prozeduren OPTSAVE und OPTLOAD. PROC OPTSAVE speichert die aktuellen Einstellungen der Systemoptionen als SAS Tabelle (data set) oder als SAS Registry Eintrag.

```

PROC OPTSAVE OUT=sasuser.myopts;
RUN;

```

Mit PROC OPTLOAD können die so gespeicherten Optionen entsprechend wieder eingesetzt werden.

Nachteil dieser an sich sehr schönen Funktionalität ist, dass längst nicht alle Systemoptionen erfasst werden! Dies zeigt sich, wenn man die Dictionary Table DICTIONARY.OPTIONS bzw. den entsprechenden SQL View SASHELP.VOPTION verwendet und deren Inhalt in einer SAS Tabelle speichert<sup>2</sup> oder sich mit PROC OPTIONS alle Systemoptionen im Log anzeigen lässt.

```

PROC SQL;
  CREATE TABLE work.myopts AS
  SELECT *
  FROM DICTIONARY.OPTIONS;

```

---

<sup>2</sup>Siehe auch Tipps & Tricks der 5. KSFE 2000 in Gießen.

Diese Tabelle enthält wesentlich mehr Optionen als die mit PROC OPTSAVE erstellte, sowie zu jeder Option noch Detailinformation (Scope, Description). Der Nachteil dieser Methode besteht darin, dass die Einstellungen nicht als Ganzes wiederhergestellt werden können, da die Dictionary Tables schreibgeschützt sind. Vielmehr muss man dazu die normale Options Anweisung verwenden und die Settings vorher in Makrovariablen speichern.

**Tipp:**

Als dritte Methode bietet sich die Funktion **GETOPTION()** an, die besonders für die Speicherung in Makrovariablen geeignet ist und außerdem den Vorteil besitzt, neben den Systemoptionen auch Grafikoptionen (goptions) auslesen zu können.

**Beispiel 3.1:**

Einstellung der Systemoptionen NOTES, SOURCE und YEARCUTOFF speichern, ändern und ursprüngliche Einstellung wiederherstellen

```
* aktuelles Setting der Systemoptionen in Makrovariablen speichern;
%LET opt1=%SYSFUNC(GETOPTION(NOTES));
%LET opt2=%SYSFUNC(GETOPTION(SOURCE));
%LET opt3=%SYSFUNC(GETOPTION(YEARCUTOFF));
* Einstellung der Systemoptionen ändern;
OPTIONS NONOTES NOSOURCE YEARCUTOFF=1900; ...
* ursprüngliches Setting der Optionen wiederherstellen;
OPTIONS &opt1 &opt2 YEARCUTOFF=&opt3;
```

Angenommen in der ursprünglichen Einstellung waren Notes und Source eingeschaltet und Yearcutoff hatte den Wert 1920, so lautet die letzte Options Anweisung nach der Auflösung der Makrovariablen:

```
OPTIONS NOTES SOURCE YEARCUTOFF=1920;
```

Mit Hilfe der **KEYWORD Option** der GETOPTION() Funktion wird die abgefragte Option mit ihrem Namen so wiedergegeben, dass sie direkt in der OPTIONS oder GOPTIONS Anweisung einsetzbar ist.

**Beispiel 3.2:**

Verschiedene Grafikoptionen mit GETOPTION() und der KEYWORD Option abfragen

```
%PUT %SYSFUNC(GETOPTION(NOTES, KEYWORD));
%PUT %SYSFUNC(GETOPTION(YEARCUTOFF, KEYWORD));
%PUT %SYSFUNC(GETOPTION(GUNIT, KEYWORD));
%PUT %SYSFUNC(GETOPTION(COLORS, KEYWORD));
```

Kann beispielsweise folgende Ausgabe im Log liefern:

```
NOTES
YEARCUTOFF=1920
GUNIT=CELLS
COLORS=( )
```

## 4 Externe DLLs in SAS verwenden (nur Windows)

DLLs (Dynamic Link Libraries) sind unter Windows ausführbare Dateien, die eine oder mehrere Routinen enthalten, welche in einer Programmiersprache wie z. B. C, C++, Pascal (Delphi), Visual Basic, etc. geschrieben worden sind. Mit SAS unter Windows ist es möglich, DLLs innerhalb eines Data Steps, der SCL oder IML aufzurufen und damit auf externe d.h. SAS-fremde Routinen zuzugreifen. Dies ist seit SAS Version 6.11 möglich.

Insbesondere stehen dadurch die Windows DLLs mit ihren API Routinen (application programming interface) zur Nutzung in SAS zur Verfügung, wodurch sich völlig neue Möglichkeiten in der Programmierung ergeben. Am Beispiel der KERNEL32.DLL von Windows 95/98/Me, NT/2000/XP soll dies an zwei darin enthaltenen Routinen gezeigt werden.

### Vorgehensweise:

1. Textdatei oder Katalogeintrag vom Typ 'Source' mit der Beschreibung der zu verwendenden Routinen mit den Anweisungen **ROUTINE** und **ARG** erstellen („Attribute Table“) und über den Fileref **SASCBTBL** referenzieren.

Entscheidend dafür ist, dass man die Routine mit ihren Argumenten kennt. Microsoft stellt die Beschreibung seiner Windows API Routinen über das Software Development Kit (SDK) im Internet zur Verfügung (siehe Literaturhinweise).

2. Über eine Funktion der MODULExy Familie aus einem Data Step, der SCL oder IML kann man auf die gewünschte Routine zugreifen. Die richtige Funktion hängt vom Typ des Rückgabewertes der Routine ab.

Data Step oder SCL:

MODULE (kein Rückgabewert), MODULEN (num.),  
MODULEC (char.),

IML:

MODULEI (kein Rückgabewert), MODULEIN (num.),  
MODULEIC (char.)

### Beispiel 4.1:

Auslesen des Pfades für temporäre Dateien mittels der Routine  
GetTempPathA der KERNEL32.DLL

(Dieser Pfad steht auch in der Windows Umgebungsvariablen TEMP)

### Windows Beschreibung der Routine:

```
DWORD GetTempPathA(  
    DWORD nBufferLength, // size of buffer  
    LPTSTR lpBuffer      // path buffer  
);
```

### SAS Beschreibung der Routine:

Als Textdatei mit beliebigem Namen speichern und mit Fileref SASCBTBL verknüpfen. Dies kann beispielsweise mit dem folgenden CARDS4 Datenschnitt automatisiert geschehen.

```
FILENAME SASCBTBL "C:\Temp\DLLAttributeTable.txt";  
  
* Definition der Attribute Table mit der Beschreibung  
  der Routinen der DLL;  
DATA _NULL_;  
  FILE SASCBTBL;  
  INPUT;  
  PUT _INFILE_;  
CARDS4;  
ROUTINE GetTempPathA  
  MINARG=2  
  MAXARG=2  
  MODULE=kernel32  
  STACKPOP=CALLED  
  RETURNS=LONG;  
ARG 1 INPUT BYVALUE FORMAT=PIB4.;  
  * DWORD nBufferLength, // size of buffer;  
ARG 2 UPDATE          FORMAT=$260.;  
  * LPTSTR lpBuffer, // path buffer (max. 254 );  
;;;;  
RUN;
```

### Anwendung der Routine im Data Step

```
DATA _NULL_;  
  LENGTH TempPfad $260;  
  rc=MODULEN("GetTempPathA", 260, TempPfad);  
  PUT _ALL_;  
RUN;
```

### Ergebnis<sup>3</sup>

```
NOTE: Variable TempPfad is uninitialized.  
TempPfad=C:\TEMP\ rc=8 _ERROR_=0 _N_=1
```

---

<sup>3</sup>Diese Note weist nicht auf einen Fehler hin! Sie lässt sich z.B. dadurch unterdrücken, dass die Variable, die durch die Routine gefüllt wird, vorher in einer RETAIN Anweisung initialisiert wird z.B. RETAIN TempPfad " ";



**Beispiel 4.2:**

Auslesen des freien Speicherplatzes eines Laufwerks in Bytes mit der Routine GetDiskFreeSpaceExA der KERNEL32.DLL

**Windows Beschreibung der Routine:**

```

BOOL GetDiskFreeSpaceExA(
    LPCTSTR lpDirectoryName, // directory name
    PULARGE_INTEGER lpFreeBytesAvailable, // bytes available
                                        to caller
    PULARGE_INTEGER lpTotalNumberOfBytes, // bytes on disk
    PULARGE_INTEGER lpTotalNumberOfFreeBytes // free bytes on disk
);

```

**SAS Beschreibung der Routine:**

Als Textdatei mit beliebigem Namen speichern und mit Fileref SASCBTBL verknüpfen.

```

ROUTINE GetDiskFreeSpaceExA
    MINARG=4
    MAXARG=4
    MODULE=kernel32
    STACKPOP=CALLED
    RETURNS=LONG;
ARG 1 CHAR INPUT          FORMAT=$CSTR260.;
    * LPCTSTR lpDirectoryName, // directory name;
ARG 2 NUM OUTPUT          FORMAT=PIB8.;
    * PULARGE_INTEGER lpFreeBytesAvailable, // bytes avail.;
ARG 3 NUM OUTPUT          FORMAT=PIB8.;
    * PULARGE_INTEGER lpTotalNumberOfBytes, // bytes on disk;
ARG 4 NUM OUTPUT          FORMAT=PIB8.;
    * PULARGE_INTEGER lpTotalNumberOfFreeBytes // free bytes;

```

**Anwendung der Routine im Data Step**

```

DATA _NULL_;
    Dir="C:\"; * oder UNC Pfad mit "\\Server\Share\";
    RETAIN verfuegbar gesamt frei 0;
    * verhindert die 'uninitialized' Note;
    rc=MODULEN("GetDiskFreeSpaceExA", Dir, verfuegbar,
               gesamt, frei);
    ProzentFrei=100*verfuegbar/gesamt;
    FORMAT ProzentFrei 5.1;
    FORMAT verfuegbar gesamt frei COMMAX15.;
    PUT _ALL_;
RUN;

```

## Ergebnis

```
Dir=C:\ verfügbar=76.349.440 gesamt=2.105.344.000
    frei=76.349.440 rc=1 ProzentFrei=3.6 _ERROR_=0 _N_=1
NOTE: DATA statement used:
    real time          0.08 seconds
```

## Literaturhinweise:

- SAS Companion for the Microsoft Windows Environment, Version 8, Chapter 13: Accessing external DLLs from the SAS System
- The Microsoft® Windows application programming interface (Windows API) in Microsoft® Platform Software Development Kit (SDK) for Windows 95/98/Me, NT/2000/XP  
<http://msdn.microsoft.com/library/> ⇒ Windows Development ⇒ Win32 API ⇒ SDK Documentation ⇒ Windows API

## 5 SAS Notes oder Wie helfe ich mir selbst?

Bevor den Anwender beim Arbeiten mit SAS die Verzweiflung packt, kann er auf verschiedene Hilfsmittel zugreifen, die ihn vielleicht/hoffentlich der Lösung näher bringen. SAS Notes und weitere Quellen im weltweiten Internet werden im folgenden vorgestellt.

Die meisten Anwender suchen den oder die Fehler zunächst bei sich selbst. Erst nach längerem Ausprobieren wagt man an die Möglichkeit zu denken, dass die SAS Software selbst das Problem darstellen könnte. Bevor man sich dann aber gleich ans Telefon hängt, um mit einem der freundlichen, aber oft auch unerfahrenen Hotline-Mitarbeiter von SAS Institute zu telefonieren, kann man zunächst die SAS Notes auf dem WWW-Server von SAS Institute ([www.sas.com](http://www.sas.com)) zu Rate ziehen. Man kann sich von der Homepage zum Technical Support „durchhangeln“ oder ruft direkt

[www.sas.com/service/techsup/search/index.html](http://www.sas.com/service/techsup/search/index.html)

auf. Hiermit gelangt man zur Suchmaschine über alle Seiten des Technical Supports inklusive der SAS Notes. SAS Notes, früher auch SAS Usage Notes, dokumentieren

- Bekannte Probleme, mit eventuell vorhandenen Fixes oder Auswegen
- Tipps und Gebrauchsanweisungen
- Fehler in der Dokumentation und zugehörige Korrekturen
- Kompatibilitätsfragen

SAS Notes werden unterschieden nach den V6 SAS Notes für die Versionen 6.06 bis 6.12 und SN SAS Notes für die höheren Versionen 7 und 8.

**Beispiel 5.?: Übertragung von SAS-Dateien mit Formaten nach SPSS**  
Wir rufen dazu oben angegebene Suchfunktion auf und geben folgende Stichworte an „SPSS data file formats“.

Die Suche liefert 17883 Dokumente, wovon die ersten 500 sortiert nach der Relevanz (in %) angezeigt werden.


Glücklicherweise erscheint gleich unter den ersten 25 Resultaten die Lösung zu unserem Problem, SAS Note SN-002899 (s.u.).

Bevor man ansonsten die Fülle von Dokumenten durchsucht, könnte man die

Suchbegriffe weiter spezifizieren, z.B. durch Angabe der tatsächlich aufgetretenen Fehlermeldung.

Die SAS Note SN-002899 enthält eine kurze Beschreibung des Sachverhalts, die Fehlermeldung, die von SPSS angezeigt wird und die Lösung, wie man das Problem umgehen kann. Am Ende erscheint eine Liste der SAS Versionen, bei denen dieses Problem beobachtet wurde und ob es für das Problem ein Fix gibt.

[TS Home](#) | [Intro to Services](#) | [News and Info](#) | [Contact TS](#) | [Site Map](#) | [FAQ](#) | [Feedback](#)


SN-002899  
Transferring user-written formats to SPSS results in errors

---

In order to transfer user-written formats from SAS to SPSS, SPSS instructs their customers to use PROC FORMAT with the CNTLOUT= option to create a control data set. This control data set is then put in SAS transport format using the XPORT engine. If the control data set is created using the V7 or V8 engine, problems are encountered when the data set is later read into SPSS. The error generated by SPSS is:

ERROR: SAS format file observation header record not found.

To circumvent the problem, create the control data set in Version 6 format by using the V6 engine on the LIBNAME statement. For example,

```
libname v6lib v6 'SAS library specification';

proc format lib=library cntlout=v6lib.fmt;
  select fmt; /* Select the name of the format */
run;
```

---

**Product:** Base SAS  
**Component:** COPY procedure  
**Priority:** N/A  
**Status:** Usage Issue  
**Date:** Fri, 25 Aug 2000

System	Version Reported	Version Fixed
AIX/6000	7 TSP1	
AIX/6000	8 TSM0	

## Anmerkungen

1. Vergleicht man den Inhalt der Note SN 002899 mit den Darlegungen von Lochmann und Ortseifen (2002) hinsichtlich dem Datenaustausch zwischen SAS und SPSS, bemerkt man, dass in der SAS Note ein entscheidender Hinweis auf die SPSS-Version fehlt, bei der dieses Problem auftritt.

2. Wird ein Problem gefixt, kann dies entweder in einer der nächsten Version geschehen oder SAS stellt auf dem Server ein Hotfix zur Verfügung, das der Anwender als selbstentpackende Datei vom SAS Server übertragen und in die bestehende SAS Software einbinden kann. Eine Liste der verfügbaren Hotfixes findet man unter <http://ftp.sas.com/techsup/download/hotfix/hotfix.html>
3. Die Veränderungen in den SAS Notes werden von SAS Institute täglich per eMail über die allgemein zugängliche Liste TSSASNOTES-L verschickt.

### Weitere Hilfen im Internet

Auf dem *WWW-Server* von SAS Institute gibt es neben den SAS Notes noch zahlreiche weitere Hilfen, z.B. technische Tipps, zusätzliche Dokumentation u.a.m.

Unabhängig von SAS Institute existieren *Diskussionslisten* und Diskussionsforen, die dem SAS Anwender hilfreich sein können.

- SAS-L, [www.sas-l.com](http://www.sas-l.com), internationale Diskussionsliste
- SAS-EDU, [www.urz.uni-heidelberg.de/statistik/sas-ah/](http://www.urz.uni-heidelberg.de/statistik/sas-ah/), deutschsprachige Diskussionsliste
- [www.belnet.com/newsgroups/sas/](http://www.belnet.com/newsgroups/sas/), deutschsprachiges Diskussionsforum der Fa. Dibera.

Und schließlich sei noch auf das *SAS Anwenderhandbuch im Netz* hingewiesen, das deutschsprachige Anleitungen und Links zu weiteren WWW-Seiten zur SAS Software darbietet.

([www.urz.uni-heidelberg.de/statistik/sas-ah](http://www.urz.uni-heidelberg.de/statistik/sas-ah))

### Literaturhinweise

- Lochmann, T. und Ortseifen, C. (2002): Datenaustausch zwischen SAS und SPSS. Proceedingsband 6. KSFE 2002, Dortmund.

## 6 PROC UNIVARIATE in V8

Die Prozedur UNIVARIATE ist in der Version 8 ein richtiger Tausendsassa geworden. Sie hat viele Eigenschaften der Prozedur CAPABILITY aus dem SAS Modul SAS/QC geerbt, was die Anwender schon lange gewünscht (oder auch bemängelt) haben. Mit dem Output Delivery System (ODS) kann man die Tabellen und Grafiken bequem in ein RTF-File überführen, das sich ohne Schwierigkeiten in Word öffnen lässt.

Im folgenden Beitrag werden einige neue Features und Statistiken der Prozedur vorgestellt, hochauflösende Grafiken gezeigt und der Export nach Word an einem Beispiel veranschaulicht.

### Neue Anweisungen

Analog zur Prozedur MEANS kennt die Prozedur UNIVARIATE in der Version die Anweisung CLASS. Damit entfällt das für BY notwendige Sortieren der Datei, und es können vergleichende Grafiken angefertigt werden (s. Beispiel 6.1). Die Anweisung INSET erlaubt dem Anwender, einzelne Kennwerte in die Grafik aufzunehmen (s. Beispiel 6.1).

### Hochauflösende Grafiken

Während mit der Option PLOT der Anweisung PROC UNIVARIATE bislang nur niedrigauflösende Grafiken (Boxplot, Stamm- und Blattdiagramm, Normal Probability Plot) im Output-Fenster erzeugt werden konnten, können ab der Version 8 – sofern SAS/GRAPH auch lizenziert ist – hochauflösende Grafiken angefertigt werden: Mit der

- Anweisung HISTOGRAM Balkendiagramme mit überlagerten parametrischen und nicht parametrischen Dichteschätzern,
- Anweisung PROBLOT Probability Plots zum Vergleich der Variablenwerte mit Perzentilen einer theoretischen Verteilung und
- Anweisung QQPLOT Quantil-Quantil-Plots.

Hochauflösende Boxplots können mit der eigenen Prozedur BOXPLOT erzeugt werden, die hier aber nicht Thema ist.

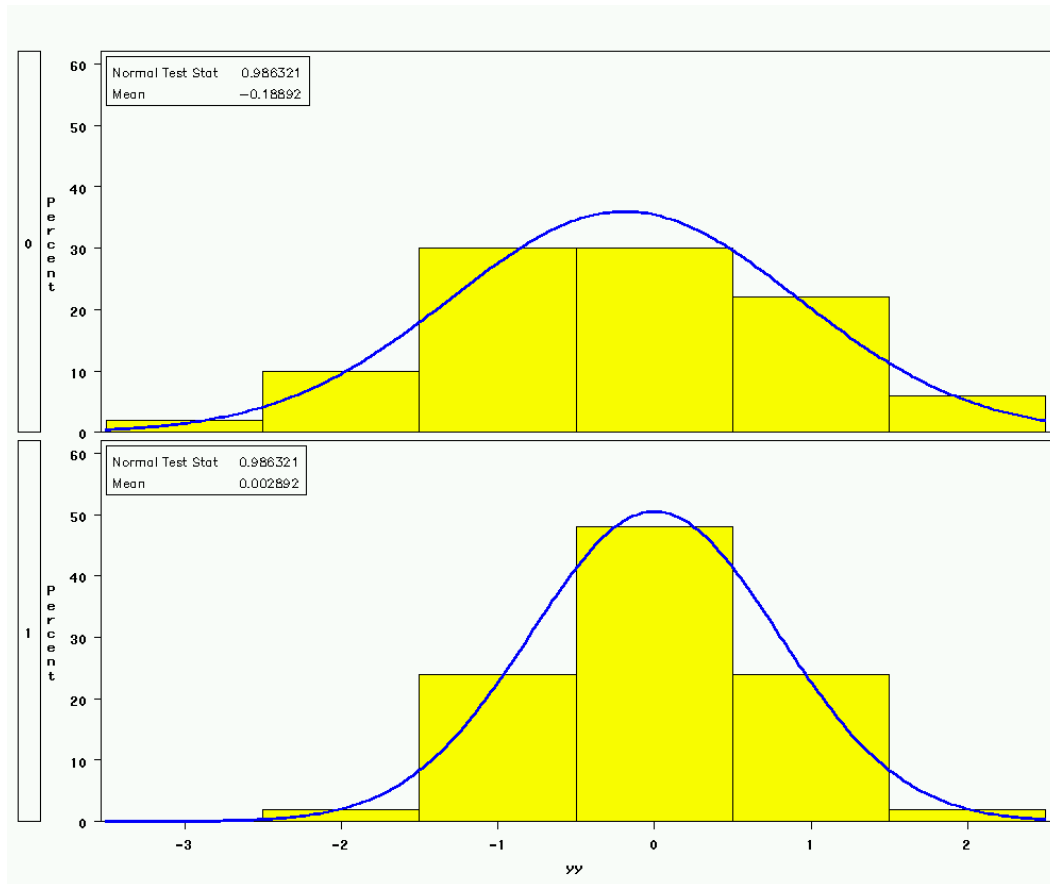
### Beispiel 6.1:

Die Daten für den nachfolgenden Prozeduraufruf werden zunächst künstlich mithilfe der Zufallszahlenfunktion NORMAL erzeugt. Die Variable yy enthält die normalverteilten Werte, die Variable gruppe wird später als Gruppierungsvariable dienen. Die Daten werden in der Datei \_a abgelegt.

```
DATA _a ;
  DO i = 1 TO 100 ;
    yy=NORMAL(123) ;
    gruppe=MOD(i,2) ;
    OUTPUT ;
  END ;
  DROP I;
RUN ;

PROC UNIVARIATE DATA=_a normal ;
  CLASS gruppe ;
  HISTOGRAM yy / NORMAL(W=3) CFILL=YELLOW ;
  INSET NORMALTEST MEAN ;
RUN ;
```

Der Prozedurschritt erzeugt neben den üblichen Kennwerten folgendes vergleichende Histogramm mit überlagerter Normalverteilungsdichtefunktion.



### Neue Statistiken (Optionen)

Auch hinsichtlich der Kennwerte hat sich bei der Prozedur UNIVARIATE einiges getan: Es können verschiedene Konfidenzgrenzen und robuste Maße berechnet sowie Lokationstest nicht nur gegen die Nullhypothese  $\mu_0 = 0$  durchgeführt werden. Im einzelnen sind dies die Optionen:

CIBASIC	Konfidenzgrenzen für Mittelwert, Standardabweichung und Varianz
CIPCTLNORMAL	Konfidenzgrenzen für Quantile, basierend auf der Normalverteilung
CIPCTLDF	Konfidenzgrenzen für Quantile, verteilungsfreie Berechnung
ROBUSTSCALE	Robuste Skalenparameter
TRIMMED	Robuste Lageparameter (getrimmte Mittelwerte)
WINSORIZED	Robuste Lageparameter (winsorisierte Mittelwerte)
MU0=	Erwartungswert unter der Nullhypothese
LOCCOUNT	Berechnung der Anzahl der Werte unterhalb des, oberhalb des und gleich dem Lokationsparameter

## Grafiken nach Word exportieren – ODS RTF

Was nützen die schönsten Grafiken, wenn man sie nicht nach Word (oder ein anderes Textverarbeitungssystem) übertragen kann. Musste man mit älteren SAS Versionen noch CGM-Files erzeugen, um diese in guter Qualität in Word darstellen zu können, kann man nun ODS RTF verwenden.

### Beispiel 6.2:

Der Prozedurschritt aus Beispiel 6.1 wird wiederholt, die Grafik und die Kennwerte allerdings mittels zweier zusätzlicher Zeilen in das RTF-File `univar.rtf` überführt. Die erste ODS RTF Anweisung definiert mittels der Option `BODY` den Namen des RTF-Files. Die zweite ODS RTF Anweisung schließt diesen RTF-File, so dass er anschließend sofort mit Word geöffnet werden kann.

```
ODS RTF BODY='c:\test\univar.rtf';

PROC UNIVARIATE DATA=_a NORMAL ;
  CLASS gruppe ;
  HISTOGRAM yy / NORMAL(W=3) CFILL=YELLOW ;
  INSET NORMALTEST MEAN ;
RUN ;

ODS RTF CLOSE;
```

Zusätzliche ODS SELECT Anweisungen erlauben, gezielte Kennwerte in das RTF-File einzutragen.

```
ODS RTF BODY='c:\test\univar.rtf';
ODS SELECT MOMENTS TESTSFORNORMALITY;

PROC UNIVARIATE DATA=_a NORMAL ;
  CLASS gruppe ;
  HISTOGRAM yy / NORMAL(W=3) CFILL=YELLOW ;
  INSET NORMALTEST MEAN ;
RUN ;

ODS RTF CLOSE;
```

Neben der Grafik erscheinen dann für die beiden Gruppen 0 und 1 je zwei Tabellen im RTF-File, wobei hier nur Gruppe 0 dargestellt wird.

Moments			
<b>N</b>	50	<b>Sum Weights</b>	50
<b>Mean</b>	-0.1889237	<b>Sum Observations</b>	-9.4461853
<b>Std Deviation</b>	1.10886681	<b>Variance</b>	1.2295856
<b>Skewness</b>	-0.178735	<b>Kurtosis</b>	-0.295297
<b>Uncorrected SS</b>	62.0343026	<b>Corrected SS</b>	60.2496943
<b>Coeff Variation</b>	-586.93895	<b>Std Error Mean</b>	0.15681745



Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.98055	Pr < W	0.5754
Kolmogorov-Smirnov	D	0.084091	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.057504	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.352988	Pr > A-Sq	>0.2500

**Quelle**

Prozedur UNIVARIATE. Online Dokumentation im HTML-Format, SAS Inst.

## 7 Die neue Fensterumgebung in V8

Mit der Version 8 ist die Fensterumgebung von SAS um einige Fenster erweitert worden. Von besonderer Bedeutung sind der SAS-Explorer und der neue – allerdings nur unter Windows verfügbare – Enhanced Editor.

### Der Enhanced Editor

Der neue Editor erlaubt keine Zeilenkommandos mehr, obwohl auch hier noch eine Zeilennummerierung zugeschaltet werden kann. Dafür bietet er neben einer farblichen Darstellung von Programm-Code noch eine ganze Reihe weiterer sinnvoller Ergänzungen. Sämtliche Optionen des Enhanced Editors werden über das Menü Tools ⇒ Options ⇒ Enhanced Editor eingestellt (dafür muss das Editor-Fenster aktiv sein). Unter dem ersten Tabber, „General“, wird u.a. eingestellt, wie die Darstellungen der einzelnen Programmschritte erfolgt (collapsible code sections). Der Editor kann Trennlinien zwischen den Schritten und/oder Klammern in der linken Spalte anzeigen. Für Freunde der F4-Taste gibt es die Checkbox „clear text on submit“, denn der Enhanced Editor löscht submittierten Text nicht mehr.

### Anpassen der Farbschema

Der mit „Appearance“ beschriftete Tabber ist für das Farbschema zuständig. Hier gibt es eine Reihe vordefinierter Schemata, die angepasst und unter neuen Namen gespeichert werden können. Der Enhanced Editor bietet Schemata für Base-Programme, Scl und HTML an. Dabei ist der Editor in der Lage, bestimmte Schlagwörter oder Abschnitte zu erkennen und farbig hervorzuheben, z.B. die Begriffe „DATA“ und „PROC“. Das gilt aber auch für Kommentare und alphanumerische Konstanten. Wer also dazu tendiert, Anführungszeichen zu vergessen, der sollte in der Auswahlliste „File elements“ den Punkt „String constant“ markieren und hier eine möglichst deutliche Farbgebung wählen. Dabei erkennt der Enhanced Editor nicht alle in SAS vorhandenen Schlüsselwörter, es gibt aber im Tabber „General“ die Möglichkeit über den Button „User defined Keywords“ solche Begriffe nachträglich aufzunehmen. Sie werden über das gleichnamige „file element“ im Tabber „Appearance“ verwaltet.

Übrigens unterstützt auch der alte Editor jetzt ein Farbcodierungs-Schema. Es kann über Options  $\Rightarrow$  Scheme angepasst werden und steht auch unter Unix, Linux und OS/390 zur Verfügung. Nachteil ist, dass der eingegebene Code erst nach Drücken der Enter-Taste farbig codiert wird, während der Enhanced Editor auf jede Eingabe direkt reagieren kann. Für den alten Editor lässt sich das color coding mit dem Kommando „SYNCOLOR <ON|OFF>“ ein- bzw. ausschalten.

### **Auto-Text**

Ein weiterer Vorteil des neuen Editors ist die Möglichkeit, Textbausteine anzulegen. Das ist z.B. sinnvoll, um in einem Projekt einen als Standard definierten Kommentarblock an den Anfang jedes Programms zu stellen. Statt diesen in einer extra Datei zu speichern, wird er als Auto-Text angelegt. Mit Hilfe einer speziellen Kombination lässt er sich einfach in ein Programm einfügen. Ich empfehle folgende Vorgehensweise:

1. Schreiben Sie den gewünschten Textbaustein in den Editor und kopieren Sie ihn dann (Strg + c).
2. Rufen Sie über das Menü „Tools“ den Punkt „Add abbreviation“ auf. Hier sehen Sie zwei Eingabefelder, in dem oberen wird ein Kürzel für den Textbaustein angegeben, das zweite Feld nimmt den einzufügenden Text auf (Strg + v).

Jetzt kann der Textbaustein verwendet werden. Wenn Sie das Kürzel in den Editor eingeben, bekommen Sie den Textbaustein angezeigt, mit der Tabulator-Taste wird er eingefügt.

## **Nützliche Tastenkombinationen**

**Anmerkung:** Die Tastenkombinationen funktionieren nach einer Installation nicht immer sofort. Dann müssen sie nochmals zugewiesen werden. Das geht wieder über das Menü des Enhanced Editors: Wählen Sie Tools  $\Rightarrow$  Options  $\Rightarrow$  Enhanced Editor Keys. Hier finden Sie in der Kategorie „Navigation“ die Punkte „Move cursor to matching brace/parent“ bzw. „Move cursor to matching DO/END keyword“. Über den Button „Assign keys...“ können Sie die Tastenkombination vergeben.

### Orientierung im Programm

Strg + G	Goto-Fenster; springt in die angegebene Zeile
Alt + Zehnerblock + bzw. -	Ein- und Ausklappen des aktuellen Programmabschnittes
Strg + Alt + Zehnerblock + bzw. -	Ein- und Ausklappen aller Programmabschnitte
Strg + F2	Setzt bzw. löscht eine Bookmark (= Markierung einer Zeile)
(Shift) + F2	Springt zur nächsten (vorherigen) Bookmark

### Bearbeiten von Text

Doppelklick auf Wort	Markiert ein Wort
Strg + u	Markierung wird in Großbuchstaben umgesetzt
Strg + l	Markierung wird in Kleinbuchstaben umgesetzt
Markieren + Tab	Ein markierter Block wird eingerückt, wenn die Tabulator-Taste gedrückt wird. Mit Shift + Tabulator wird der Block nach links gerückt.

### Hilfe bei der Fehlersuche

Strg + 7	Erzeugt Kommentarzeichen um einen markierten Block; Mit Strg + Shift + 7 werden die Kommentarmarken wieder entfernt
Strg + 9	Sucht eine passende Klammer; kann sehr nützlich sein, wenn mehrere Funktionen ineinander verschachtelt sind
Alt + 9	Sucht passende Do/End-Kombinationen

### Der SAS-Explorer

Der SAS-Explorer ist die neue Schaltstelle in einer SAS-Session. Er zeigt alle verfügbaren Ressourcen (libraries, filerefs) an und erleichtert die Übersicht wesentlich. Dabei ersetzt er im Prinzip folgende Fenster: Libname, Filename, Catalog und Var, die aber natürlich nach wie vor zur Verfügung stehen (wenn auch in überarbeiteter Form). Der Explorer steht übrigens auch unter OS/390 zur Verfügung, hier muss er aber mit dem Kommando „explorer“ explizit gestartet werden.

Es werden (unter dem Menüpunkt „view“) verschiedene Anzeigemodi angeboten:

- Large icons
- Small icons
- List
- Details

Über den Menüpunkt View  $\Rightarrow$  Show tree kann die eigentliche Explorer-Sicht ein- oder ausgeschaltet werden. Ist der Punkt aktiviert, wird das Fenster in zwei Bereiche aufgeteilt. Links befindet sich eine Baumansicht der SAS-Umgebung, d.h. alle allokierten Libraries und Filenames, mit einem Doppelklick lassen sich Libraries und Kataloge öffnen. Im rechten Fensterbereich wird der Inhalt des im linken Bereich ausgewählten Items angezeigt.

Für jeden Item gibt es ein Standard-Verhalten das ausgeführt wird, wenn auf den Item ein Doppelklick erfolgt: z.B. Wird eine Library auf- oder zugeklappt, eine Tabelle wird im Viewtable geöffnet usw. Außerdem verfügt der Explorer über item-spezifische Kontextmenüs.

Interessant wird das ganze jetzt, weil die Kontextmenüs im Explorer bearbeitet und erweitert werden können. Z.B. wird eine Tabelle nach einem Doppelklick zwar im Viewtable geöffnet, aber nur im Browse-Modus. Außerdem zeigt der Viewtable zunächst einmal die Variablen-Label an, nicht die Namen. SAS-Programme, die in Entries vom Typ SOURCE gespeichert werden, können über das Kontextmenü auch nicht einfach wieder im Enhanced Editor bearbeitet werden, sie werden nämlich im Notepad-Editor geöffnet.

Um ein Kontextmenü des Explorers zu bearbeiten, müssen Sie wie folgt vorgehen.

1. Wählen Sie (bei aktivem Explorer) die Menüpunkte Tools  $\Rightarrow$  Options  $\Rightarrow$  Explorer.
2. Im oberen Bereich sehen Sie eine Liste, für welche unterschiedlichen Bereiche Einstellungen vorgenommen werden können. Um die Einstellungen für Tabellen zu ändern, muss die Kategorie Members ausgewählt sein, für Katalog-Einträge entsprechend „Catalog Entries“.
3. Dann können Sie in der großen Liste „Table“ bzw. „SOURCE“ auswählen.
4. Wenn Sie jetzt den Button „Edit...“ anwählen, erscheint ein Fenster in dem die Belegung des Kontextmenüs angezeigt wird.
5. Bei Betrachten der bereits vorhandenen Menüpunkte fallen zwei Dinge auf, erstens, hinter den Kontextmenüs verstecken sich einfache SAS-Kommandos, und zweitens, die aktuelle Markierung wird in einer Reihe

von Platzhaltern übergeben. So steht hinter dem Kontextmenü „open“ für die Tabelle das Kommando `VIEWTABLE %8b.'%s'.DATA`, für den SOURCE-Entry steht hier `NOTEPAD %8b.%32b.%32b.%8b.`

(Der Ordnung halber, %8b. und %32b. stehen für acht bzw. 32 Zeichen lange Zeichenketten, '%s' für eine Zeichenkette in Anführungszeichen; Sie werden dies aber kaum brauchen, da Sie i.d.R. die gleichen Parameter verwenden werden wie die normalen Kontextmenüs).

6. Das Bearbeiten oder Hinzufügen neuer Unterpunkte für das Kontextmenü ist denkbar einfach. Zum Bearbeiten wählen Sie den Punkt aus und klicken dann auf den Button „Edit...“, zum Hinzufügen auf „Add...“.

### Beispiel 7.1:

Das Kontextmenü für SAS-Tabellen wird um den Menüpunkt „Bearbeiten“ ergänzt, hier wird die Tabelle im Edit-Modus geöffnet und zeigt gleich die Variablenamen an.

Sie wählen dafür „Add...“ und geben unter „Action“ den Text „&Bearbeiten“ ( das „&“ markiert das Tastaturkürzel) an, unter „Action Command“ den Befehl `VIEWTABLE %8b.'%s'.DATA openmode=edit;names;`

### Beispiel 7.2:

SOURCE-Entries sollen im Enhanced Editor geöffnet werden. Wählen Sie den Eintrag „&Open“ aus und dann „Edit...“. Geben Sie als Kommando `WHOSTEDIT;COPY %8b.%32b.%32b.%8b.` ein.

Wenn Sie die Änderungen jeweils mit OK bestätigt haben, stehen die neuen Menüpunkte/Verhaltensweisen zur Verfügung.

### Quelle:

SAS Help

SAS Online Documentation SAS Institute

